

51CTO.com

技术成就梦想

我们只谈开发

# 开发月刊

Development Monthly

2011年10月

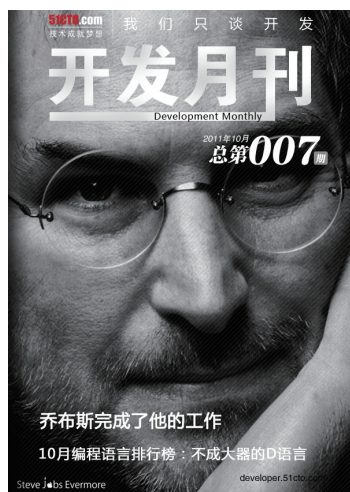
总第007期

乔布斯完成了他的工作

10月编程语言排行榜：不成大器的D语言

Steve Jobs Evermore

developer.51cto.com



## 编程排行 Billboard

- 3 十月编程语言排行榜：不成大器的D语言

## 特别报道 Special News

- 5 乔布斯完成了他的工作

## 专题报道 《李洋谈Android安全专栏》

- 6 李洋谈Android系统及程序授权机制  
10 李洋谈Android应用程序管理  
12 李洋:Android设备短信丢失技术原因  
13 李洋谈Android系统启动过程剖析

## 技术热点 Techlogy hot

- 14 论编程语言的可读性  
16 程序员是自己创业还是去放高利贷?  
17 9月Java备忘录Google专利谈判破裂  
19 浅谈C#结构  
21 开发者需了解过去闻所未闻的新技能  
23 9月Web技术最前沿:jQuery成版本帝  
25 Dart之于JavaScript正如C#之于C++  
27 JavaScript大辩论:改进还是彻底放弃  
29 微软能否撑起Silverlight的明天?  
30 数据库迁移之何去何从

## 特别推荐 Best topic

- 32 HTML 5对阵Flash



# 10月编程排行榜:不成大器的D语言

2011年10月10日 Tiobe 发布了 2011年10月的编程语言排行榜。本期排行榜我们看到 Lua 这门语言蹿升很快,而我们今天要介绍的不那么成气候的 D 语言已然跌出前 20 名。

今天 Tiobe 发布了 2011年10月编程语言排行榜,本期排行榜我们看到 Lua 这门语言蹿升很快,而传统的竞争对手 C# 和 PHP 正在交替自己的位置。Assembly (汇编语言)正在大踏步的向后退,从 2011年3月份的中游位置一路滑坡到本次排行榜的第 20 位,这门每个中国程序员大学必修课似乎正在衰落。

下面是前 20 名的编程语言排行榜情况:

Position Oct 2011	Position Oct 2010	Delta in Position	Programming Language	Ratings Oct 2011	Delta Oct 2010
1	1	=	Java	17.913%	-0.25%
2	2	=	C	17.707%	+0.53%
3	3	=	C++	9.072%	-0.73%
4	4	=	PHP	6.818%	-1.51%
5	6	↑	C#	6.723%	+1.76%
6	8	↑↑	Objective-C	6.245%	+2.54%
7	5	↓↓	(Visual) Basic	4.549%	-1.10%
8	7	↓	Python	3.944%	-0.92%
9	9	=	Perl	2.432%	+0.12%
10	11	↑	JavaScript	2.191%	+0.53%
11	10	↓	Ruby	1.526%	-0.41%
12	12	=	Delphi/Object Pascal	1.104%	-0.45%
13	13	=	Lisp	1.031%	-0.05%
14	14	=	Transact-SQL	0.909%	+0.09%
15	23	↑↑↑↑↑↑↑	PL/SQL	0.903%	+0.30%
16	24	↑↑↑↑↑↑↑	Lua	0.802%	+0.25%
17	16	↓	RPG (OS/400)	0.757%	+0.05%
18	15	↓↓↓	Pascal	0.721%	-0.05%
19	-	=	Assembly*	0.622%	-
20	17	↓↓↓	Ada	0.609%	-0.09%

## 雄心壮志在我胸——D 语言

D 语言从诞生的那一刻, Digital Mars 公司就

希望 D 语言能取代如日中天的 C 语言。对! C 语言系列将被 D 语言替代,包括 C 和 C++,还有 C# 和 Object-C。这样的雄心壮志不可谓不大。

总体来说 D 语言是一种通用的系统和应用编程语言。它是比 C++ 更高级的语言,同时还保持了生成高效代码以及直接访问操作系统 API 和硬件的能力。

D 语言不是脚本语言,也不是一种解释型语言。它不需要虚拟机、宗教、或者高于一切的哲学。它是给实际的编程者使用的实际语言,它帮助编程者快速、可靠的完成易维护、易于理解的代码。

在 2007 年,51CTO 开发频道还发布过一篇文章,标题就是《D 语言能否成为明日之星》。可惜的是,4 年过去了,这个预言似乎还没有实现。

附: D 语言的 Hello World 代码

```
import std.c.stdio;

int main(char[][] args)

{ printf("Hello World!\n"); return 0; }
```

## D 语言为什么会出现?

C 和 C++ 诞生已经几十年,随着程序员们研究的日益深入和兼容性的要求,这两门语言也日渐臃肿。举个例子, C 标准将近 500 页, C++ 标准大概有 750 页。这样庞大规模的语言,想继续创新显然工作量太大了。

程序员们还有一个怪癖,就是自己熟用的一

## 10 月编程语言排行榜: 不成大器的 D 语言 II

些方法或者某些特性,就会一直使用下去。即使未来有更好的方法,也不会再想去学习。渐渐的,使用编程语言的“孤岛”效应就产生了,造成代码通常在编译器之间是可移植的,但在程序员之间移植就不那么容易了。

所以 D 语言的创造者希望能让大家过渡到 D 语言,这样使代码易于在编译器之间、在机器之间、在操作系统之间移植。同时要同本机的 C 语言应用程序二进制接口相兼容。对于熟悉 C 语言或者 C++ 的人来说,学习曲线要短。

综上所述, D 语言真的希望自己能帮到各位程序员,有一个更加轻松的编译环境。

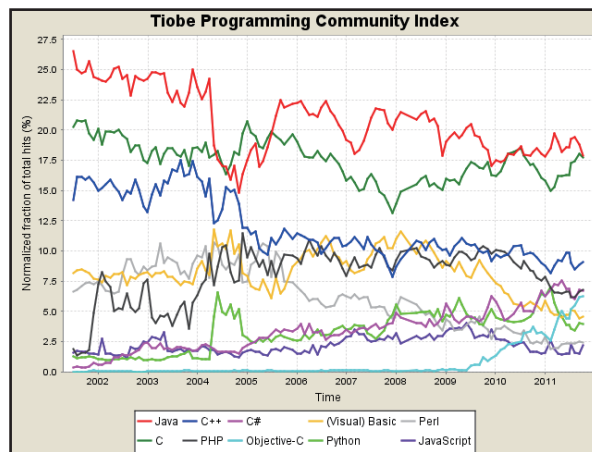
### D 语言目前不成大器的原因

在勉强迈进编程语言排行榜前 20 之后,很多程序员都表示没听说过 D 语言,这种尴尬可能印证了 D 语言目前难成大器的近况。没有大公司联盟的推广,这样的语言注定是悲哀的。C 和 C++ 正处于如日中天的地步,大型企业级软件和众多游戏开发都采用它们,现在想换掉 C 和 C++ 无异于让这些开发企业徒增成本而已。

开发语言的寿命是很长的,计算机硬件在 60 多年里已经发生了翻天覆地的变化,价格也越来越便宜。但现在的程序员还依旧使用 40 年前甚至更古老的语言进行开发,即使是年轻的 Java 也进入中年。由此可见 D 语言想打翻身仗,还需要谋求更多的大企业来支持,让更多的程序员能用上 D 语言是当务之急。

下面是本期编程语言排行榜的其他排名数据和趋势走向。

前 10 名 编 程 语 言 走 势 图



### 下面是第 50 到 100 的编程语言排名

(Visual) FoxPro, ABC, Algol, Bash, bc, BETA, BlitzMax, Bourne shell, CFML, cg, CL (OS/400), Clean, cT, Dylan, Eiffel, Factor, Free Pascal, Groovy, Icon, IDL, Io, J, JavaFX Script, JScript.NET, Korn shell, LabVIEW, Lingo, LPC, MAD, Magic, Max/MSP, MOO, MUMPS, NATURAL, Oberon, Occam, OpenCL, OpenEdge ABL, Oz, PostScript, PowerShell, REXX, S, SPSS, SuperCollider, VBScript, VHDL, X10, xBase, Z shell

### 下面给出了编程语言类别的一年变化趋势

Category	Ratings Oct 2011	Delta Oct 2010
Object-Oriented Languages	55.9%	+0.7%
Procedural Languages	38.0%	-1.5%
Functional Languages	4.1%	+0.4%
Logical Languages	2.0%	+0.4%

Category	Ratings Oct 2011	Delta Oct 2010
Statically Typed Languages	64.6%	+1.7%
Dynamically Typed Languages	35.4%	-1.7%

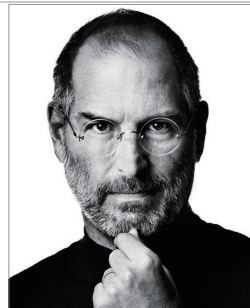
更多内容,请查看原文,链接:

<http://developer.51cto.com/art/201110/295939.htm>

作者: 彭凡 ■

# 乔布斯完成了他的工作

美国最近发生了两件大事,第一, They lost jobs. (他们失业了!); 第二, They lost Jobs. (Steve Jobs 死了! )。乔布斯(以下简称 Jobs )完成了他的工作,Jobs 完成了他的 jobs。



这个世界是不公的,越是天才越受嫉妒,包括上天对其的眷恋也是少之又少。

乔帮主在有生之年一统了江湖,但却没有千秋万代。他的创意带给人们快乐,他的离开带给人们悲伤,久病缠身的他终于可以休息了,不属于这个时代的 Jobs 终于完成了他的 jobs,谢谢你, Steve Jobs。

## 不了解 Jobs: 但又得说点什么

这个世界上的多数谁也不认识谁,道听途说的居多。作为一个国内媒体从业者,我对 Jobs 的了解仅限于其他媒体的报道,相信大多数人都是如此。现在,在新闻、微博上,大家都在疯狂缅怀他,有点像谁不说点什么仿佛就跟不上时代、不新潮了似的,也许这就是某些人使用 Jobs 产品的初衷吧。虽然我很老土,但我也不能免俗,因为这事儿是我的 jobs (工作),媒体多是这样。在 Jobs 辞职时写过一篇文章(推荐阅读:《移动互联网》第一季终于 hold 住了),本以为今年 IT 业界最大的事莫过于此,但看来我错了。

在全世界的苹果专卖店门前,“果粉”对其离去的缅怀方式几乎大同小异,其实真正缅怀的是苹果产品曾经带给我们的美好回忆。但是,知晓开启我们这种快乐感觉大门的人走了,以后的生活是否继续美好,变得前途未卜。有时我们会带着情绪问自己,为什么要去了解一个外国人?! 他

了解我们么?! 很遗憾,是的。

## 了解 Jobs: 诠释“传奇”

我们如何说了解一个人呢? 大体上对其说的话,做的事有所了解就行了。至于人家的隐私、探秘,你不了解反而倒是正常现象。我敢说,如果不是苹果 iphone、ipad 街机一般的存在,如果不是现在苹果公司如日中天的业绩,没几个人知道 Jobs。但满足了这两样背景的人大有人在,我们对他们有多了解呢?

但我们真的很了解 Jobs,了解他传奇般的产品,了解他传奇般的经历,了解他及他的苹果许许多多……因为,他了解需求,了解市场,了解我们。他的产品总是抓住了我们内心最最本质的需求,很奇怪,他真的了解我们!

就消费者而言,在一个很精准的搞懂其需求的人面前,消费者总会花很多时间了解这个神奇的家伙,Jobs 就是这么个人。就媒体而言,也愿意把闪光灯更多的对向他。我们关注 Jobs 的时间会比一般读者略长,总结的内容会多一些,这是我们的 jobs (工作),我们需要向读者诠释“传奇”的细节。总体上,对 Jobs 的了解胜过了我们对身边人的了解。

## 不喜欢 Jobs: 不喜欢神

对于他的竞争对手而言,Jobs 创造的一个又一个奇迹令他们无所适从。

## 乔布斯完成了他的工作 II

无论是 MP3、手机还是平板电脑,苹果在 Jobs 的带领下,将一个又一个的行业冲击地体无完肤,整个行业都把苹果以及 Jobs 当做了假想敌,他受不受欢迎可想而知。

除了对手,使用对手产品的消费者大多也不喜欢 Jobs。随着苹果硬件、软件、开发环境的日趋成熟,没有人愿意承认自己和苹果同价位产品的选择是错误的,比如谷粉、诺粉们。但不可否认,他们自己对于自己手里的产品的厌恶之情或许更甚于对 Jobs 以及苹果的反感。

我也不喜欢 Jobs,我不信世界上有如此完美的人,我不愿承认传奇人物的存在,总是在探寻他的不足达到自我心理平衡,我是个凡人,不喜欢神。

### 喜欢 Jobs: 快乐的理念

这个世界不是每个人都喜欢你,就像不是每个人都喜欢自己的工作一样,Jobs 同理。但奇怪的是,老乔手下的人对于自己的工作却多是乐在其中。

在苹果早期,Jobs 麾下的 Macintosh 团队就是一支快乐的“海盗团队”。

每个新加入 Macintosh 团队的员工,都会领到一件海盗 T 恤衫,上面印着:“做海盗! 不做海军!” 海盗和海军的譬喻,是 Jobs 本人的发明。他那个时候常说的两句口头禅是:

“当海盗比当海军更快乐。”

“能当海盗,为啥还要当海军?”

找到杰出的人,组建不受教条约束的快乐团队,是其手下喜欢 jobs (工作),喜欢 Jobs 本人的原因。这种快乐的理念不仅影响了其下的团队,

也传递给了消费者。人们对产品的关注不再仅仅是性能、外观与价格,人们选择苹果成了一种快乐的体验。

没有 Jobs 的传奇表演,就不会有快乐的苹果产品体验。人们应该庆幸自己和他生活在同一时代,有幸让自己体会到久违的快乐。

### 结语

说真的,除了 Jobs 的家人,除了多事的媒体,再过一月,谁还会想起这位传奇人物离去的日期。人对他人的想念多仅限于某段时间,人对悲伤的自愈能力或者是遗忘能力是超强的。下次再见到 Jobs 的新闻应该就在科技媒体的年终总结上了,下下次……我就不再敢想了,也许会出现下一个 Jobs? 人们还会想起这个家伙么?

在当下,人们脑海中的 Jobs 大多骨瘦如柴却神情坚毅,也许这也将是我对 Jobs 的最终记忆。其实,对于乔布斯,人们能记住他们的两三款成功产品——iphone、nano、ipad 就行了,就像比尔盖茨和微软桌面,宫本茂和红白机……包括牛顿和苹果。如果你真的在某个领域为世界作出了贡献,人们处于感恩,会记住你的,这已是对你最大的褒奖。

你曾说过:“如果你把每一天都当做生命中最后一天去生活的话,那么有一天你会发现你是正确的”。

谢谢,Jobs,这次你又对了。(作者/景琦) ■







移动开发正在成为软件领域的发展趋势,51CTO 非常荣幸邀请到从事计算机网络信息安全研发工作 10 多年的安全专家李洋为 51CTO 移动开发频道开设的专家专栏,在本期《开发月刊》这个专栏里,李洋老师将向我们介绍“Android 安全框架”、“Android 系统及应用程序授权机制”等内容,希望读者喜欢。

Android 相关专题:

Android 组件——Service 详解

<http://mobile.51cto.com/mobile/android/service/>

51CTO 移动开发频道寄语

# 李洋谈Android系统及程序授权机制

作者 / 李洋

Android 开发包工具将代码和资源数据文件编译成以 .apk 后缀的安装文件。单个 .apk 文件就是一个应用程序,可被用来安装在 Android 系统的手机上。但是了解 Android 系统及程序授权机制吗? 本文李洋老师将向我们介绍“Android 系统及应用程序授权机制”。

## 1.Android 系统权限定义

Android 系统在 /system/core/private/android\_filesystem\_config.h 头文件中对 Android 用户 / 用户组作了如下定义,且权限均基于该用户 / 用户组设置。>Android 系统在 /system/core/private/android\_filesystem\_config.h 头文件中对 Android 用户 / 用户组作了如下定义,且权限均基于该用户 / 用户组设置。

值得注意的是: 每个应用程序在安装到 Android 系统后,系统都会为其分配一个用户 ID, 如 app\_4、app\_11 等。 以下是 Calendar 和 Terminal 软件在 Android 系统中进程浏览的结果(其中,黑色字体标明的即为应用分配的用户 ID):

Android 系统中进程浏览的结果

USER	PID	PPID	VSZ	RSS	WCHAN	PC	NAME
app_16	2855	2363	216196	20960	ffffffff	afd0ee48 S	com.android.providers.calendar
app_91	4178	2363	218872	25076	ffffffff	afd0ee48 S	jackpal.androidterm

在 Android 系统中,上述用户 / 用户组对文件的访问遵循 Linux 系统的访问控制原则,即根据长度为 10 个字符的权限控制符来决定用户

/ 用户组对文件的访问权限。该控制符的格式遵循下列规则:

(1) 第 1 个字符: 表示一种特殊的文件类型。其中字符可为 d(表示该文件是一个目录)、b(表示该文件是一个系统设备,使用块输入 / 输出与外界交互,通常为一个磁盘)、c(表示该文件是一个系统设备,使用连续的字符输入 / 输出与外界交互,如串口和声音设备),“.”表示该文件是一个普通文件,没有特殊属性。

(2) 2 ~ 4 个字符: 用来确定文件的用户 (user) 权限;

(3) 5 ~ 7 个字符: 用来确定文件的组 (group) 权限;

(4) 8 ~ 10 个字符: 用来确定文件的其它用户 (other user,既不是文件所有者,也不是组成员的用户) 的权限。

(5) 第 2、5、8 个字符是用来控制文件的读权限的,该位字符为 r 表示允许用户、组成员或其它人可从该文件中读取数据。短线“-”则表示不允许该成员读取数据。

(6) 第 3、6、9 位的字符控制文件的写权限,该位若为 w 表示允许写,若为“-”表示不允许写。

(7) 第 4、7、10 位的字符用来控制文件的制造权限,该位若为 x 表示允许执行,若为“-”表示不允许执行。

举个例子,“drwxrwxr-- 2 root root 4096 2 月 11 10:36 guo”表示的访问控制权限(黑色字体标明)为: 因为 guo 的第 1 个位置的字符是 d。



## 李洋谈 Android 系统及程序授权机制 II

所以由此知道 guo 是一个目录。第 2 至 4 位置上的属性是 rwx, 表示用户 root 拥有权限列表显示 guo 中所有的文件、创建新文件或者删除 guo 中现有的文件, 或者将 guo 作为当前工作目录。第 5 至 7 个位置上的权限是 rwx, 表示 root 组的成员拥有和 root 一样的权限。第 8 至 10 位上的权限仅是 r--, 表示不是 root 的用户及不属于 root 组的成员只有对 guo 目录列表的权限。这些用户不能创建或者删除 guo 中的文件、执行 junk 中的可执行文件, 或者将 junk 作为他们的当前工作目录。

### 2. Android 应用程序权限申请

每个应用程序的 APK 包里面都包含有一个 AndroidManifest.xml 文件, 该文件除了罗列应用程序运行时库、运行依赖关系等之外, 还会详细地罗列出该应用程序所需的系统访问。程序员在进行应用软件开发时, 需要通过设置该文件的 uses-permission 字段来显式地向 Android 系统申请访问权限。

该文件的基本格式如下:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.
android.com/apk/res/android"
    package="cn.com.fetion.android"
    android:versionCode="1"
    android:versionName="1.0.0">
    <application android:icon="@drawable/
icon" android:label="@string/app_name">
        <activity android:name=".welcomActivity"
            android:label="@string/app_name">
            <intent-filter>
```

```
        <action android:name="android.intent.
action.MAIN" />
        <category android:name="android.intent.
category.LAUNCHER" />
    </intent-filter>
    </activity>
</application>
    <uses-permission android:name="android.
permission.SEND_SMS"></uses-permission>
</manifest>
```

如上述文件描述中加下划线的斜体部分, 该文件的作用是说明该软件需要发送短信的功能。

值得注意的是: 通过测试发现一种特殊的情况, 应用程序可以在程序运行时申请 root 权限, 如右图, 在使用 Android Terminal Emulator 时尝试使用 su 命令切换到 root 用户。若用户已通过 hacking 的方式使得 Android 系统获得了 root 权限, 则可以允许该程序以 root 用户权限执行; 反之即算用户选择“允许”, 也不能使程序以 root 用户权限执行。

### 3. Android 系统对应用程序权限申请的处理方式分析

对 Android 源代码中的如下文件进行分析:

(1) InstallAppProgress.java: 其路径为 \packages\apps\PackageInstaller\src\com\android\packageinstaller\InstallAppProgress.java;

(2) PackageInstallerActivity.java: 其路径为 \packages\apps\PackageInstaller\src\com\android\packageinstaller\PackageInstallerActivity.java;

未完查看详细:

[http://mobile.51cto.com/hot-282966\\_1.htm](http://mobile.51cto.com/hot-282966_1.htm)

# 李洋谈Android应用程序管理

作者 / 李洋

## 1. 简介

Android 一词的本义指“机器人”，谷歌的基于 Linux 平台的开源手机操作系统以“Android”命名，以机器人为标志。Android 系统早期由谷歌开发，后由开放手持设备联盟 (Open Handset Alliance, OHA) 开发。OHA 由谷歌与 34 家厂商一起于 2007 年 11 月 5 日创立，包括手机制造商、电信运营商和芯片制造商，如下表。OHA 创立的同时，谷歌发布了 Android 系统。

Android 采用了软件堆层 (Software stack, 又名以软件叠层) 的架构，主要分为三部分，包括底层 Linux 内核、Android 中间件和系统关键应用程序。Android 中间件以 Dalvik 虚拟机为基础构成，这是由谷歌自己设计开发的 Java Virtual Machine。Android 上的应用使用 Java 语言开发，Android SDK 为在 Android 系统上开发 Java 应用提供了一整套开发工具和 API。

## 2. 应用程序组成

Android 的每个应用程序都对应一个安装包，称之为 APK (Android Package)，在系统运行时每一个 APK 对应一个操作系统进程。

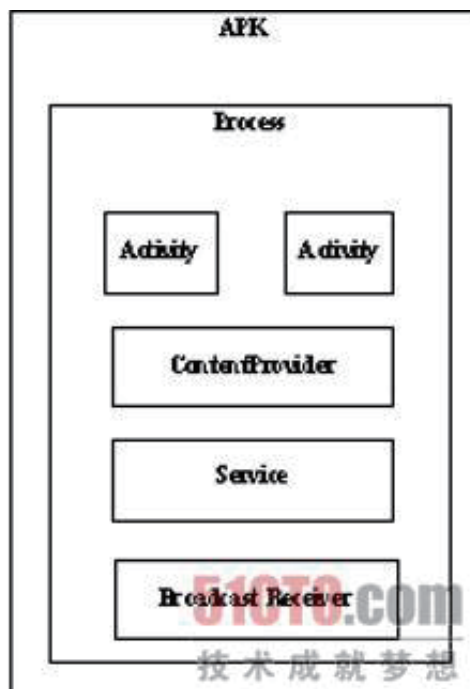
(1)Activity: Activity 就是一个界面，这个界面里面可以放置各种控件，如短信编辑框等；

(2)Service: 服务是运行在后台的功能模块。如文件下载、音乐播放程序等；

(3)Content Provider: 它是 Android 平台应用程序间数据共享的一种标准接口，它以类似于 URI(Universal Resources Identification) 的方式来

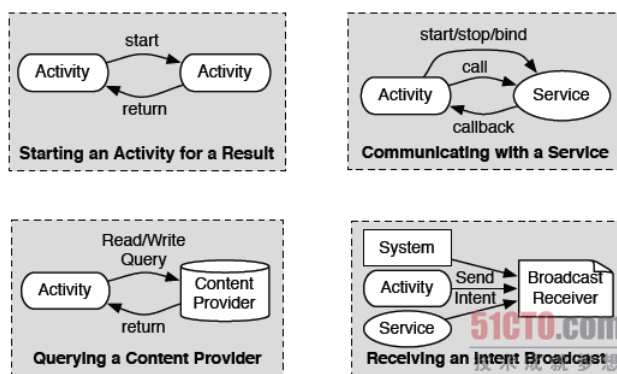
表示数据，如：content://contacts/people/1101；

下图所示为 APK 中所包含的组件



(4)Broadcast Receiver: 与此组件相关的概念是 Intent, Intent 是一个对动作和行为的抽象描述，负责一个程序的组件之间、不同程序之间进行消息传递，Intent 包括目标组件地址和消息数据。Broadcast Receiver 组件负责实现侦听 Intent 消息并对其作出反应。

下图为 APK 中各组件之间的典型关系示例。



## 李洋谈 Android 应用程序管理 II

## 3. 应用程序生命周期管理

Android 的每个应用程序以一个 Linux 进程运行,默认情况下,在进程中执行一个单一的线程,应用程序所有的组件均在这个进程的线程中运行。当然也可以为任意进程衍生出其它线程。Android 系统会尽可能长的延续一个应用程序进程执行,确保其正常执行结束。

除了应用程序执行结束后,正常释放内存外,在内存过低超过阈值时,系统仍然会不可避免需要移除旧的进程。为决定保留或移除一个进程,Android 将每个进程都放入一个“重要性层次”中,依据则是它其中运行着的组件及其状态。重要性最低的进程首先被消灭,然后是较低的,依此类推。重要性共分五层,依据重要性列表如下:

(1) 前台进程是用户操作所必须的,任一时间下,仅有少数进程会处于前台,仅当内存实在无法供给它们维持同时运行时才会被杀死。

(2) 可视进程没有前台组件,但仍可被用户在屏幕上所见。可视进程依然被视为是很重要的,非到不杀死它们便无法维持前台进程运行时,才会被杀死。

(3) 服务进程是由 `startService()` 方法启动的服务,它不会变成上述两类。尽管服务进程不会直接为用户所见,但它们一般都在做着用户所关心的事情(比如在后台播放 mp3 或者从网上下载东西)。所以系统会尽量维持它们的运行,除非系统内存不足以维持前台进程和可视进程的运行需要。

(4) 背景进程包含目前不为用户所见的 activity。这些进程与用户体验没有直接的联系,

可以在任意时间被杀死以回收内存供前台进程、可视进程以及服务进程使用。

(5) 空进程不包含任何活动应用程序组件。这种进程存在的唯一原因是作为缓存以改善组件再次于其中运行时的启动时间。系统经常会杀死这种进程以保持进程缓存和系统内核缓存之间的平衡。

Android 会依据进程中当前活跃组件的重要程度来尽可能高的估量一个进程的级别。比如说,如果一个进程中同时有一个服务和一个可视的 activity,则进程会被判定为可视进程,而不是服务进程。

此外,一个进程的级别可能会由于其它进程依赖于它而升高。一个为其它进程提供服务的进程级别永远高于使用它服务的进程。

## 4. 实现应用程序开机自动启动

任何一个 Android 应用程序要实现开机自动启动,必须做到如下两点:

1) 继承 `BroadcastReceiver` 类,并实现 `onReceive()` 方法,在此方法中启动应用程序。

2) 在应用程序的 `AndroidManifest.xml` 文件中订阅 `BOOT_COMPLETED` 消息,如下:

```
<receiver android:name="HelloWorldStartup
IntentReceiver">
    <intent-filter> <action android:
name="android.intent.action.
BOOT_COMPLETED" />
    </intent-filter>
</receiver>
```

本文全文完■



# 李洋:Android设备短信丢失技术原因

作者 / 李洋

很多 Android 手机用户都有手机短信莫名其妙的丢失的情况。很多人都不明白,为什么会丢失呢?本文将探究一下个中原因,希望能对用户有所帮助。

## 第一类原因:存储原因

### 1.1 Android 管理的 SQLite 数据库出错

Android 对短信的管理都是通过 SQLite 来进行的,且所有短信都是存储在相应的数据库中,其存储路径通常为 dbdata/databases/com.android.providers.telephony/(不同的 Android 系统,路径可能有细微区别)下的 mmssms.db 数据库,该数据库维护了近 20 张表格,其中短信就存储在 sms 表格中。因此,若该数据库出错或者被删除(手机没有 root 权限无法读取和删除该数据库),则所有的短信将会丢失。

### 1.2 存储介质出现故障

Android 手机存储(Flash 存储)是短信默认的物理存储介质,若手机存储出现故障,则可能使得短信丢失。同样道理,若将短信设定为存储在 SD 卡上,若 SD 卡出现故障,同样可能造成短信丢失。

## 第二类原因:应用程序原因

### 2.1 具有操作短信权限的应用程序出错

操作短信的 Android 应用程序较多,如 Go 短信、超级短信等都可以辅助用户进行短信操作,因此对存储在手机存储介质上的短信也有删除权限。若这些软件不可预期(不由用户执行)地执行短信删除的操作,则短信也可能丢失。

## 第三类原因:Android 系统的稳定性问题

有数以万计的搜索结果显示 Android 手机具有不稳定性,如手机意外重启后短信丢失等。这都是由于 Android 系统自身的稳定性造成的。用户可以通过升级最新的 Android 版本,以及定期备份短信和数据的办法来避免由于系统不稳定而造成的数据丢失问题。

## 第四类原因:配置及操作原因

### 4.1 手机短信删除设置

Android 手机可以对手机短信进行设置,并在一次会话(短信发送方-接收方进行的短信交流称为会话)超过设定的阈值,则系统将会删除会话中保留时间最长的短信,以保证不超过阈值。经过测试,这种删除不是批量的,是一条一条的。如当会话的短信数目达到阈值 200 时,用户发送第 201 条短信,则 Android 自动将该会话的第 1 条短信删除。具体的操作方法是:信息->设置->存储设置->设置删除旧短信、短信限制和彩信限制。

### 4.2 恢复出厂设置

用户使用 Android 手机的恢复出厂设置功能也可能造成短信、应用程序和其他数据的丢失。具体操作方法是:应用程序->设置->隐私权->恢复出厂设置。

### 4.3 误操作

Android 支持短信的批量删除,若用户误操作,也有可能将短信意外地全部删除。具体操作方法是:信息->设置->删除会话。

# 李洋谈Android系统启动过程剖析

作者 / 李洋

李洋老师在前文中向我们介绍了“Android 安全框架”和“Android 系统及程序授权机制”。李洋老师将继续介绍 Android 系统架构的介绍——《Android 系统启动过程剖析》。

## 1. 系统启动流程简介

在 Linux 内核启动后, init 1(1 号进程) 将作为第一个用户空间 (Linux 虚拟内存的大小为 232( 在 32 位的 x86 机器上 ), 内核将这 4G 字节的空间分为两部分。最高的 1G 字节供内核使用, 称为“内核空间”。而较低的 3G 字节供各个进程使用, 称为“用户空间”。) 的进程来启动 Android 系统, 该启动流程可以分为如下 5 个阶段, 如下图:

点击下面链接查看原图:

<http://mobile.51cto.com/hot-285155.htm>

(1) 启动准备: 该阶段包括创建文件系统的基本目录、打开基本输入、输出设备, 初始化日志功能等;

(2) 解析 init.rc 文件: 该阶段对 init.rc 脚本文件进行解析, 主要对 Service( 服务 ) 和 Action( 动作 ) 进行解析。其中, Service 由命令 (Command) 和一系列服务的附加内容 (Option, 选项) 组成, 如: “service vold /system/bin/vold” 为一个 Service, 而 “socket vold stream 0660 root mount” 则为配合该服务使用的 Option; Action 则由一系列的命令组成, 如: “on init mkdir /system” 为系统初始化时建立系统文件夹的 Action;

(3) 触发需要执行的 action: Action 需要在

Triggers( 触发条件 ) 中调用, 本阶段对需要执行的 Action 进行触发, 并根据触发条件将需要执行的 Action 放入 Action 队列;

(4) 执行在 action 队列中的命令: 对上一阶段触发的 Action 以及 Service 进行执行。并在此过程中, 派生了 Zygote 和 Service Manager 两个非常重要的进程;

(5) 循环处理事件: init 进程进入无限循环, 处理设备插入 / 拔出, 服务属性状态变化和 signal 事件等。

## 2. 源码分析结果

对 android-2.3.3\_r1 版本中的如下源码文件进行分析:

(1) init.c: 路径为 system/core/init/init.c

(2) init\_parser.c: 路径为 system/core/init/init\_parser.c

(3) builtins.c: 路径为 system/core/init/builtins.c

(4) property\_service.c: 路径为 system/core/init/property\_service.c

(5) keycords.c: 路径为 system/core/init/keycords.c

(6) signal\_handler.c: 路径为 system/core/init/signal\_handler.c

总结得出系统启动流程对应的源代码文件及函数如下 (注: 以下函数间的顺序执行关系使用 “>” 表示; 函数间的调用执行关系使用 “à” 表示)

## 李洋谈 Android 系统启动过程剖析 II

## 2.1 第一阶段 (启动准备)

具体的函数执行过程如下:

```
mkdir > mount > open_devnull_stdio >  
log_init
```

## 2.2 第二阶段 (解析 init.rc 文件)

具体的函数调用过程如下:

```
init_parse_config_file à parse_config à  
parse_new_section à parse_service ( 或者  
parse_action) -> parse_line_service( 或者  
parse_line_action)
```

## 2.3 第三阶段 (触发需要执行的 action)

具体的调用过程如下:

```
action_for_each_trigger("b  
oot", action_add_queue_tail); à  
action_add_queue_tail ( class_start default) à  
action_remove_queue_head à do_class_start
```

## 2.4 第四阶段 (执行在 action 队列中的命令)

具体的调用过程如下:

```
execute_one_command à  
action_remove_queue_head à do_class_start  
à service_for_each_class à service_start_if_not_  
disabled à service_start
```

## 2.5 第五阶段 (循环处理)

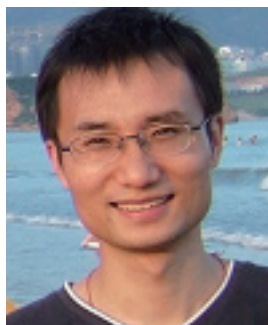
具体的循环处理过程如下:

```
for (;){  
poll > handle_property_set_fd >  
handle_keychord > handle_signal  
}
```

未完查看详细:

<http://mobile.51cto.com/hot-285155.htm> ■

## ■ 关于作者



李洋

李洋博士毕业于  
中科院计算所

10 多年来一直从事计算机网络信息安全研发工作,曾主持和参与多项国家重点项目以及信息安全系统和企业信息安全系统的研发工作。具有 Linux 系统应用、管理、安全及内核的研发经验,擅长网络安全技术、协议分析、Linux 系统安全技术、Linux 系统及网络管理、Linux 内核开发。

## ■ HTML 5 开发小 TIP

## 1. 新的 Doctype

尽管使用 <!DOCTYPE html>,即使浏览器不懂这句话也会按照标准模式去渲染。

## 2. 去掉 link 和 script 标签里面的 type 属性。

## 3. 加 / 不加 括号

HTML5 没有严格的要求属性必须加引号,闭合不闭合,但是建议加上引号和闭合标签。

4. 让你的内容可编辑,只需要加一个 contenteditable 属性。

## 5. Local Storage

使用 Local Storage 可以永久存储大的数据片段在客户端(除非主动删除),目前大部分浏览器已经支持,在使用之前可以检测一下

window.localStorage 是否存在



# 论编程语言的可读性

## ■ 前言

新的语言在这个互联网时代层出不穷,比如从 Go 语言到 Scala。似乎 C++、COBOL 这样的老语言,就应该扔进历史垃圾堆了。其实不然。

最近我在网上看到一些人把各种脚本语言放到一起进行比较。脚本语言: PHP, Perl, Python, Ruby。我的第一反应,以及第二反应是:讨厌! 我有我自己的偏好,从没向任何人说过。我喜欢可读性好的程序,程序里出现的任何的特殊字符(任何非数字、字母的字符)都会影响我的阅读速度。这让我思考起我们应该如何设计编程语言? 我们语言设计是在进步还是在倒退还是在停步不前?

回到我刚开始写程序的时候,大概是 40 年前,那时的三大编程语言是 FORTRAN、COBOL 以及后来的 BASIC。FORTRAN(FORMula TRANSlation 的简写)被数学家(我的数学专业的哥哥必须要学它)和科学家们使用。COBOL(其中的 B 代表 Business)被用于企业应用。BASIC 是一种教学语言,在企业里广为使用。COBOL 因为它的冗长的语法而受一些人痛恨,而同时却也被另外一些人所喜爱。但它至少是很容易理解的。看看下面循环的写法:

```
PERFORM VARYING WS-  
BOTTLE-  
NUM FROM 98 BY -1  
UNTIL WS-BOTTLE-  
NUM < 2  
END-PERFORM  
很像是一个英语句子。跟 C 语言风格的语言(C#)比较:  
  
for (WSBOTTLENUM =  
98; WSBOTTLENUM >= 2;
```

WSBOTTLENUM--){} 哪一个更显而易见? 除非你谎称自己没经验。

BASIC (这里是 Visual Basic) 某些地方可读性处于中等水平。

For WSBOTTLENUM = 98 To 1 Step -1  
Next 步长——每次减少的数——很容易理解,至少我是这么认为。让我们看一些非常简单的东西。

```
k = i / 10;
```

这会让初学者抓狂。这是做什么的? 我们这些程序员当然会知道,但对于初学者,这种指令操作方式会让他们困惑。你可以把它与相同功能的 COBOL 语言比较:

divide i by 10 giving k 有点长? 不错,但一个初学者能看懂。我现在并不是在主张重新回到 COBOL 语言,诚然,相比起当时在纸片上打孔的时代,现代的集成开发工具和像 Intellisense 这样的功能让我们的编程变的容易的多了。可是,我还是要说,一些入门语言应该罗嗦一些——英语句子比“#号在这种编程语言里代表什么”要容易上手的多。

只是好玩,如果你想看一看各式各样的语言,可以访问 99 Bottles of Beer 这个网站。

这个网站收集了用各种不同语言编写的 99 Bottles of Beer 的歌词。目前已经有 1434 种不同的编程语言和语言变种来表现这首歌。■

## ■ 编者按

各位程序员,你们想去创业的目的是什么呢? 挣钱养家糊口还是实现自己的技术理想? 相信一百个人有一百个想法,其实我们可能还不如去放高利贷。

# 程序员是自己创业还是去放高利贷?

本文标题不是危言耸听,在中国不断紧缩银根的今天,很多在北京工作的白领将自己的积蓄拿到鄂尔多斯或者温州放高利贷。高于银行几倍甚至十几倍的利率,让越来越多的人卷入了高利贷的狂潮。中国高利贷年息 60% 到 150%,看来我们程序员自己创业还真不如去放高利贷。

中国正在扼杀中国青年的创造力

经济学家郎咸平说,高房价正在扼杀中国的创造力。“对于一个国家来说,整个创造力的下降其实才是最大的危机。你看 10 年前,马云花 50 万创立了阿里巴巴,小马哥的这个腾讯也是差不多 50 万,陈天桥也是 50 万,他们都用了差不多 50 万创立了自己的企业。再看看现在的年轻人,如果手里有 50 万的话,他们肯定不会投到创业上去,那干嘛呢? 肯定是拿这 50 万去付首付了。”这番话在 2011 年上半年之前是有效的。而在 2011 年,投资房地产已经让位给了高利贷。

下面我们来看一位北漂白领的个人经历。

你可能就是中国未来的“高布赛克”

《高利贷者》是巴尔扎克的另一部使人印象深入的中、短篇小说,书中描写了一个叫高布赛克的高利贷者。而在现在的中国,各位白领和技术人员正成为高布赛克的预备军。

月薪 8000 元的小胡北漂已经 10 年,手里攥着 10 万元的积蓄,没赶上股市大涨,也没赶上炒

房,如今想想不能再错过家乡的这一轮‘发财机会’。这所谓的机会是什么呢?

在今年中秋节的鄂尔多斯老乡会上,当小胡听到大家都心潮澎湃地谈论着鄂尔多斯的民间借贷,动了心弦。“2 分月息,10 万元一年的利息就是 2.4 万元,相当于我 3 个月不工作就拿了工资。有一个老乡表示自己有渠道,并能保证收贷人的资金实力,我和几个同乡也就跟着加入进去了。”

这样的例子已在蔓延,下面我们再看 51CTO 刚刚报道两位美国程序员创业的艰辛例子。

上面两位仁兄,2007 年夏天,放弃了稳定收入的工作后,非常非常辛苦的创业。到 2010 年二月,持续将近 2 年半没有任何收入。而他们又做出了什么能向人展示的成果吗? 没有。

直到 2010 年 2 月份,距离他们开始创业 2 年半之后,我们又产生了另外一个想法:收集那些正在兴起但发展迅速的日常交易信息。这个想法来的很及时,正合他们的特长。因为最近 9 个月里他们一直在做本地交易信息收集。

只用了 3 天时间,一切全变了。3 天的仓促编码,他们上线了这个新功能,做了一些简单宣传,用户十分喜欢这个功能,4 个月后,吃惊的投资者向我们注入了 1 百万美元。一年之后,他们募集到了 6 百万美元。更多内容请访问原文:

<http://developer.51cto.com/art/201109/295109.htm> ■

## ■ 编者按

51CTO 九月 Java 备忘录将记录本月有关 Java 的点点滴滴,本月 Java 业界比较重大的事件非 Google 与甲骨文的专利谈判破裂。

# 9月Java备忘录Google专利谈判破裂

美国时间 9 月 19 日,谷歌 CEO 佩奇与 Oracle CEO 拉里森坐到一起,就 Google Android 侵犯甲骨文 Java 专利进行谈判。但最终的结果是不欢而散,双方未达成任何和解。美国加州圣何塞地方法官保罗·格雷瓦尔 (Paul Grewal) 要求两家公司在当地时间周三 (21 日) 进行第二轮和解谈判,地点定于加州圣何塞地区法院。

在 51CTO 记者看来,这样的诉讼官司将会持续很长的时间,亦如当年微软和 Sun 最终的和解一样。不到最后撕破脸皮的时候,谁都不愿意做出让步,顶多就是浪费一些律师费罢了。

鲸吞 SUN 之后的甲骨文,以继任者的身份向 Google 提出几十亿的诉求,正是看中了 Android 未来的发展前景上。从目前来看,唯一能对抗住苹果 iPhone 的,只有 Google 的绿机器人。

赔偿款不是 Oracle 诉讼 Google 的目的,那几十亿美元对于 Oracle 说多不多,说少不少。但是 Oracle 拉里森的最终目标,还是希望在 Android 市场能分得一杯羹。如果达成的和解协议里,能让 Oracle 达到这个目的,那么赔偿款的数字可以多砍砍。

Oracle 与 Google 的猫鼠游戏最终结局如何?肯定是握手言欢,各大目的。不信?那就让咱们拭目以待。

## 51CTO 技术沙龙——Java 快速开发之路

在 2011 年 9 月 3 日下午 51CTO 举办的第

13 期 51CTO 技术沙龙中,有幸请到 Java 开源平台——J-Hi 的两位创始人张昊老师和肖金华老师。二位讲师分别从理论和实战来为大家介绍 J-Hi,打开 Java 快速开发之路。



本期沙龙资料下载:

<http://down.51cto.com/data/245952>

主题:《Java 快速开发的分析与探索》(张昊)

主题:《Java 快速开发平台使用》(肖金华)

欢迎您访问此次沙龙的视频专题

本期沙龙的第一位嘉宾张昊的演讲题目是《Java 快速开发的分析与探索》,在本期沙龙报名推出时,引发了很多网友疑惑“Java 快速开发与敏捷开发有什么区别呢?”,张昊老师在沙龙中为大家解答了这个问题:

敏捷开发是一种以人为核心、迭代、循序渐进的开发方法。而 J-Hi 则是以工具为核心,利用



## 9 月 Java 备忘录: Google 与甲骨文专利谈判破裂 II

组件化、标准化来提高开发效率,节省开发时间,降低成本,二者可以相互结合。敏捷开发中的短周期迭代和需求测试似乎是解决之道,但不一定适合每一个公司,而 J-Hi 可应用到每个公司,甚至适用于每个 Java 开发者。

介绍清楚敏捷开发与快速开发的区别后,张昊老师也正式进入演讲主题《Java 快速开发的分析与探索》。

在张昊介绍 Java 快速开发的分析与探索后,他的同事也是 "J-Hi" 项目另一位发起者肖金华老师来介绍 J-Hi 平台的由来及其实现思想,通过一个实际场景,现场通过 J-Hi 做出 demo。在演示过程中详细介绍平台的各项主要功能。让大家对这有一个清楚的理解。

### 甲骨文放弃 Linux Java 许可计划

甲骨文在 2011 年 8 月 31 日已经终止了 2006 年由 Sun 创立的“操作系统 Java 分销许可”。这份收费许可允许 Linux 开发商将 Sun 的 Java 版本整合到他们的 Linux 发行版中。

Sun 是 2006 年在 JavaOne 大会上将 Java 开源后推出这一许可的,目的是在免费 OpenJDK 的开发过程中,确保用户所获得的 Linux 中包含经过全面测试的 Sun Java。

根据甲骨文自己的解释是,废除这个许可计划是因为 OpenJDK 正在逐步成熟,多数 Linux 厂商都已经选择了 OpenJDK。当前的版本已经发展到 OpenJDK 7,并且成为了甲骨文自己的 JDK 7 的基础。

### Eclipse 3.7.1 发布 支持 Java 7

Eclipse 3.7 发布于今年 6 月,那时,基于

Eclipse 严格的发布流程,Java 7 的一些特性尚处在测试或发布流程中。这也意味着,为保证 Eclipse 的按时发布,对 Java 7 特性支持的部分只好被搁置。通常,这种小版本的发布不值一提,与以往不同的是,此版本的 Eclipse JDT(Java Development Tools)中增加了对 Java 7 的支持。

下载地址:

<http://www.eclipse.org/downloads/> ■

本文其他部分,请参考原文,链接:

<http://developer.51cto.com/art/201110/295486.htm>

■ 51CTO 精品杂志推荐《Linux 运维趋势》

《Linux 运维趋势》是由 51CTO 系统频道策划、针对 Linux/Unix 系统运维人员的一份电子杂志,内容从基础的技巧心得、实际操作案例到中、高端的运维技术趋势与理念等均有覆盖。



本杂志长期处于探索期,需要更多来自大家的意见与参与。如果您对这份电子杂志感兴趣,那就请您下载阅读;想要帮助我们做的更好,请进入我们的 Linux 运维趋势技术圈留下您的宝贵意见建议。

读者讨论组: <http://g.51cto.com/linuxops/>

邮件订阅入口:

<http://os.51cto.com/art/201011/233915.htm>

投稿邮箱: [yangsai#51cto.com](mailto:yangsai#51cto.com)

发布周期: 每个月的第二个星期五

往期《Linux 运维趋势》下载汇总页:

<http://down.51cto.com/zt/71>

# 浅谈C#结构

结构是一种值类型,并且不需要堆分配。结构的实例化可以不使用 `new` 运算符。如果声明一个 10000 个 `Point` 对象组成的数组,为了引用每个对象,则需分配更多内存;这种情况下,使用结构可以节约资源。

结构是使用 `struct` 关键字定义,与类相似,都表示可以包含数据成员和函数成员的数据结构。

一般情况下,我们很少使用结构,而且很多人也并不建议使用结构,但作为 .NET Framework 一般型别系统中的一个基本架构,还是有必要了解一下的。

## 结构的特征:

结构是一种值类型,并且不需要堆分配。结构的实例化可以不使用 `new` 运算符。

在结构声明中,除非字段被声明为 `const` 或 `static`,否则无法初始化。结构类型永远不是抽象的,并且始终是隐式密封的,因此在结构声明中不允许使用 `abstract` 和 `sealed` 修饰符。

结构不能声明默认构造函数(没有参数的构造函数)或析构函数,但可以声明带参数的构造函数。结构可以实现接口,但不能从另一个结构或类继承,而且不能作为一个类的基,所有结构都直接继承自 `System.ValueType`,后者继承自 `System.Object`。结构在赋值时进行复制。将结构赋值给新变量时,将复制所有数据,并且对新副本所做的任何修改不会更改原始副本的数据。在使用值类型的集合(如 `Dictionary<string, myStruct>`)时,请务必记住这一点。结构类型的变量直接包含了该结构的数据,而类类型的变量

所包含的只是对相应数据的一个引用(被引用的数据称为“对象”)。但是结构仍可以通过 `ref` 和 `out` 参数引用方式传递给函数成员。结构可用作可以为 `null` 的类型,因而可向其赋 `null` 值。

```
structA
```

```
{publicintx; // 不能直接对其进行赋值  
publicinty; public static stringstr = null; // 静态  
变量可以初始化 publicA(intx,inty) // 带参数的构造  
函数 {this.x =x;this.y =y; Console.WriteLine  
("x={0},y={1},str={2}", x, y,str); } }classProgram  
{staticvoidMain(string[] args) { A a =newA(1,2); A  
a1 =a; a.x =10; Console.WriteLine("a1.x={0}",a1.  
x); Console.Read(); } }结果为: x=1,y=2,str=
```

```
a1.x=1
```

此时 `a1.x` 值为 1 是因为,将 `a` 赋值给 `a1` 是对值进行复制,因此, `a1` 不会受到 `a.x` 赋值得改变而改变。

但如果 `A` 是类,这时 `a` 和 `a1` 里的 `x` 引用的是同一个地址,则 `a1.x` 的值会输出 10。

结构的装箱与拆箱我们知道,一个类类型的值可以转换为 `object` 类型或由该类实现的接口类型,这只需在编译时把对应的引用当作另一个类型处理即可。

与此类似,一个 `object` 类型的值或者接口类

## 浅谈 C# 结构 II

型的值也可以被转换回类类型而不必更改相应的引用。当然,在这种情况下,需要进行运行时类型检查。由于结构不是引用类型,上述操作对结构类型是以不同的方式实现的。

当结构类型的值被转换为 object 类型或由该结构实现的接口类型时,就会执行一次装箱操作。

反之,当 object 类型的值或接口类型的值被转换回结构类型时,会执行一次拆箱操作。

与对类类型进行的相同操作相比,主要区别在于:

装箱操作会把相关的结构值复制为已被装箱的实例,而拆箱则会从已被装箱的实例中复制出一个结构值。

因此,在装箱或拆箱操作后,对“箱”外的结构进行的更改不会影响已被装箱的结构。

structProgram

```
{staticvoidMain(string[] args) {inti =1;objecto=i; // 隐式装箱 i =123; Console.WriteLine("i={0},o={1}",i,o); Console.Read(); } } // 结果为: i=123,o=1
```

结构与构造函数我们知道结构不能使用默认的构造函数,只能使用带参数的构造函数,当定义带参数的构造函数时,一定要完成结构所有字段的初始化,如果没有完成所有字段的初始化,编译时会发生错误。结构可以使用静态构造函数吗?

可以,结构的静态构造函数与类的静态构造函数所遵循的规则大体相同。

结构的静态构造函数何时将触发呢? 结构的实例成员被引用,结构的静态成员被引用,结构显示声明的构造函数被调用。但是创建结构类型

的默认值不会触发静态构造函数。

为什么结构不能自定义无参数的构造函数?

结构类型的构造函数与类的构造函数类似,用来初始化结构的成员变量,但是 struct 不能包含显式默认构造函数,

因为编译器将自动提供一个构造函数,此构造函数将结构中的每个字段初始化为默认值表中显示的默认值。

然而,只有当结构用 new 实例化时,才会调用此默认构造函数。对值类型调用默认构造函数不是必需的。

```
structA {staticA() { Console.WriteLine("I am A."); }publicvoidFun() { } }classProgram {staticvoidMain(string[] args) { A a=newA(); a.Fun(); // 结构的实例成员被引用 Console.Read(); } } 结果为: I am A.
```

结构与继承:

一个结构声明可以指定实现的接口列表,但是不能指定基类。

由于结构不支持类与结构的继承,所以结构成员的声明可访问性不能是 protected 或 protectedinternal。结构中的函数成员不能是 abstract 或 virtual,因而 override 修饰符只适用于重写从 System.ValueType 继承的方法。

在设计编程语言时将结构设计成无继承性?

其实类的继承是有相当的成本的——由于继承性,每个类需要用额外的数据空间来存储“继承图”来表示类的传承历史。

本文未完,更多内容请访问: ■

<http://developer.51cto.com/art/201109/293004.htm>



# 开发者需了解过去闻所未闻的新技能

作者 / 核子可乐译

除了编程技术之外,如今的软件开发者们还需要了解许多在过去闻所未闻的技能

如今的软件开发者们已经不必像前辈们那样担心诸多因素,尤其是编码工作中最大限度降低内存占用(即使这意味着执行时间将会大大延长)或者局域网连接最大带宽不能超过 14.4KB 每秒等等。(尽管其中有些时至今日仍然具有一定的指导意义甚至有重新发挥作用的可能,但毕竟不是主流趋势。)

不过从另一个角度来说,当下在软件开发、硬件开发、系统及网络管理以及其它 IT 专业领域充斥着大量全新的技能乃至业务范畴,这些内容在过去往往根本不存在(所谓的‘过去’涵盖很广,从三个月之前到五年、十年甚至二十年都说得通),或者只出现于预算庞大的某些特定机构或应用程序当中。

“大家可以给自己必需的技能列一张单子,其中具体内容的取舍与我们何时走出学校、对待工作是否认真以及接受新的技术、产业动态与工程发展的积极性密切相关,”Embarcadero 技术公司开发部门副总裁兼企宣负责人 David Intersimone 如是说。

无论各位是刚刚跻身于软件开发者行列(或者是正有此打算),还是已经成为一位经验丰富的专业人士,了解自己需要掌握的内容都是很有价值的。这样大家才能具备足够的竞争力,轻松自如地选择继续从事这一行业、重新选择其它领域或者着力争取晋升。

因此在这里我要与此前自己编写的“回顾那些失传的技能”一文做个呼应,着重探讨“发掘全新的技能”——相信我,无数 IT 专业人士们正紧盯这些新兴领域,因此同样身为软件开发者你绝对不能错过这个将技能添加进自己知识结构中的机会。(声明:很明显,大多数情况下我不会讨论具体某种编程语言的相关话题)

## 函数库的灵活使用

第一条冲击到我传统观念的新技能就是,开发者们对预置在新语言中大量分类库及模板库的把握能力,这一点在 Java、C++ 乃至 Python 中都表现明显。如今我们要想完成复杂的应用程序,对函数库指令加以充分运用绝对是必不可少的能力。如果我们只懂得使用某种语言而不会使用函数库的话,那么软件成果只能是空谈。

## 站在用户立场打造产品

就在两三年前,“信息架构”与“可用性工程”还属于冷僻词汇。而到了 2010 年,围绕这两个关键词产生的招聘岗位不断涌现。总结起来,这些技能其实是用于为用户打造良好的使用体验——用户界面设计、信息展示架构以及独特的平面布局,这一切已经组成了全新的正规专业知识领域。

## 异步编程及其它相关技术

将业务平台向云计算转移主要通过基于 Web 的接口实现,因此如今异步编程正越来越受到业界的关注。眼下即使是最简单的应用程序也可能会需要异步编程技术,这一切都归结于 AJAX 的广泛应用,其编程风格迥异于以往。

## 开发者需了解过去闻所未闻的新技能 II

因此开发者们恐怕要投入相当大的心力以跟上时代的脚步了。

### 扩大技术的涉猎范围

随着时代的发展,业界对技术的要求也日新月异。如今精通某个单独领域的员工已经不那么抢手,能够掌握自己专业周边的各类相关技术才足以笑傲职场。客户的要求越来越多,而那些万金油式的全面人才已然成为猎头公司中公认的热门人选。

### 特色专长造就全新岗位

在这一话题当中,我希望用客观现状代表主观评述。在下面这份来自某招聘管理者的列表中,罗列出了数条能够为求职者加分的技能内容:

1. 基于语音识别技术的非可视化用户界面设计师;
2. 无硬盘大屏幕移动设备软件设计师;
3. 位置及方向响应(类似于导航类)应用程序设计师;
4. 石油及天然气勘探及地质领域的辅助软件专家;
5. 体内嵌入式医疗器械设计专家。

通过这份列表,我们可以看出,除了软件编写技能之外,广阔的 IT 天地间还有许多特色专长的用武之地。

### 开发工作中的快速反应与协作能力

今天的开发者们需要对软件开发过程中的快速反应具备更详尽的了解。反应速度与协作能力常常能够为企业 provide 极大的市场竞争优势。驱动测试开发、自动化验收测试、代码重构等等工作内容的顺利实施都迫切需要上述能力的支持。

## 围绕可部署性、可扩展性及可管理性开展开发工作

系统管理员们比软件开发商更了解具体产品,不仅是从业务应用角度,从企业负责人的采购决断角度更是如此。因此,开发商应该从管理者的立场密切监控产品的部署情况、扩展性以及管理难度。从这一点出发,企业的竞争力将会得到大大提升,而且管理员的一句赞许永远强过投入巨资打造的广告。

### 其它建议

以下是来自专家的其它一些补充内容:

◆ 为严格要求可扩展性的云计算提供类似 Hadoop 的分布式计算技术;

◆ “大数据”分析能力,以应对网络及科学数据等带来的庞大数据量,比如说那些用户数量惊人的超大型门户网站。

◆ 为了提高生产力而部署的 Web 应用程序框架,例如 Ruby on Rails 以及 Grails 等。

◆ 动态“脚本语言”强调开发者在实时性能表现方面的生产力,Python、Ruby、PERL、JavaScript 以及 Groovy(作为 Java 的替代方案存在)等都在此列。

高级软件工程师 Amy Unruh 还补充说,“当下对于开发者们来说,了解关于实时搜索、流水线数据分析以及活动数据流管理方面的知识非常重要。此外大家还应该在自己的知识储备中时常添加一些小型体系,包括微格式的使用、服务相关 API 的支持等等,这些都是计算原则之外颇具价值的实用技能。”

诸位读者在工作过程中发现了哪些必不可少的技能,又是如何将相关学习付诸实践的呢? ■

# 9月Web技术最前沿:jQuery成版本帝

作者 / 七武海

九月迎来学生开学潮,欢度中秋,喜迎国庆,个人觉得9月是一年中最好的月份,天气好假期也挺多,最重要的有十一长假可以期待。下面我们就向你推荐九月Web技术最前沿。

## jQuery 1.6.4 正式版发布

jQuery 1.6.4 正式版发布了!此版本较jQuery 1.6.3进行了小幅改进。包括:

**Data:** 修复了用“-”所组成的数据属性名不能被采用骆驼式命名法(函数名中的每一个逻辑断点都有一个大写字母来标记)的函数所读取。

**Event:** 修复了IE8中\$(“form”).live(“submit”, fn)语句没有被<button type=“submit”>所激发的问题。

```
<script>
$(function(){
    $('form').live("submit", function(){
        alert("before submit");
    });
});
</script>
<form action="javascript:alert('submit!!!');">
    <input type="submit" value="submit by input">
    <button type="submit">submit by button</button>
</form>
```

详细: <http://developer.51cto.com/art/201109/291021.htm>

下载地址: <http://code.jquery.com/jquery-1.6.4.js>

上一个版本jQuery 1.6.3也是九月初发布的,可见jQuery现在的版本的更新是很频繁的,有网友直言可以和Google chrome相媲美了,响当当的版本帝。从另一个角度来看,这也和jQuery广

受开发者欢迎都莫大关系,从51CTO.COM关键词排名来看,已经超过HTML 5,位居开发频道关键词榜首。

## Apache Wicket 1.5 正式版发布

5个月过后,Apache Wicket终于发布了1.5正式版。Wicket是一个Java语言的Web开发框架,与Struts, WebWork, Tapestry相类似。其特点在于对Html和代码进行了有效的分离(有利于程序员和美工的配合),基于规则的配置(减少了XML等配置文件的使用),学习曲线较低(开发方式与C/S相似),更加易于调试(错误类型比较少,而且容易定位)。

详细: <http://developer.51cto.com/art/201109/290250.htm>

下载地址: <http://www.apache.org/dyn/closer.cgi/wicket/1.5.0>

## Java Web 框架 Play Framework 2.0 预览版发布

Java Web 框架 Play Framework 发布了2.0预览版!Play!是一个Rail风格的full-stack(全栈的)Java Web应用框架,采用目前Java开源界最流行的RESTful架构设计。

包括一个简单的无状态MVC模型、Hibernate的持久性对象、基于Groovy的模板引擎以及建立一个现代化网络应用所需的一切。

Play!的关键特性:

- ◆ 此框架自动编译和重新装载源文件的任何改变。
- ◆ 智能捆绑HTTP参数到Java方法参数。
- ◆ 基于Apache Mina的快速HTTP服务器。



## 9 月 Web 技术最前沿 :jQuery 成版本帝 II

◆ 一个基于 Groovy 的强大的模板引擎,具有多层继承,定制用户标签的能力等。

◆ 优秀的错误报告功能:当发生异常,此框架会直接显示出错代码,甚至是模板代码。

新版本的主要改进:

◆ 新版本集成了一个新的构建系统。

◆ 引入了强大的异步特性。

◆ 新增了原生的 Java 和 Scala 支持。

Play 2.0 目前正在紧张有序的开发中, APIs 有可能会做出改变,你可以下载预览版一饱眼福。

详情: <http://developer.51cto.com/art/201109/290477.htm>

下载地址: [Download Play-2.0-preview](#)

源码: <http://github.com/playframework/play20>

### MySQL 5.5.16 发布

MySQL 5.5.16 发布了!该版本改进了部分功能,修复了一些 Bug,如下。

功能改进包括:

◆ 扩展了 RESET SLAVE 语句,可使用 ALL 关键字。除了删除 master.info、relay-log.info 以及所有的 relay 日志文件,RESET SLAVE ALL 还可以清除所有的连接信息,否则它们会随着 RESET SLAVE 的执行而保留在内存中。

◆ 一个新的实用工具,mysql\_plugin,可以帮助管理员管理 MySQL 服务器负载。提供了可选择的方式,可在服务器启动时手动指定 --plugin-load 选项,可以在运行时使用 INSTALL PLUGIN 和 UNINSTALL PLUGIN 语句。

修复的 Bug 包括:

◆ InnoDB 存储引擎:重新启用了从 InnoDB 插件中移除的“随机预读”功能,该功能默认是关

闭的,可以通过 innodb\_random\_read\_ahead 配置选项来启用它。由于该功能在某些情况下可以提高性能,而在一些情况下降低性能,所以在设置之前,检查是否需要开启或关闭该功能。

◆ 由于 INFORMATION\_SCHEMA 查询,需要打开 .frm 或 .TRG 文件,以及扫描大量的表,“元数据锁”子系统开销增大。例如,SELECT COUNT(\*) FROM INFORMATION\_SCHEMA.TRIGGERS。

◆ 在 Mac OS X 10.7 (Lion) 中编译失败。

◆ 在 profiling 禁用或未被编译时, set\_thd\_proc\_info() 仍进行不必要的文件名长度检查。

◆ CHECK TABLE 和 REPAIR TABLE 不能找出 MERGE 表中的问题,如内部表丢失或存储引擎错误。

◆ 损坏的表映射事件的处理可能会导致服务器崩溃。

◆ 当 argc = 0 时,嵌入式服务器崩溃。

◆ 使用 RPM 包升级时会再创建一个 test 数据库,这是不可取的,尤其是当数据库管理员已经移除了 test 数据库时。

详细信息参阅: <http://dev.mysql.com/doc/refman/5.5/en/news-5-5-16.html>

下载地址: <http://dev.mysql.com/downloads/mysql/5.5.html#downloads>

开发版的 Nginx 1.1.4 发布了。

Python 3.2.2 正式版发布

详细请查看:

<http://developer.51cto.com/art/201109/293936.htm> ■

# Dart之于JavaScript正如C#之于C++

在丹麦举行的 Web 开发者大会上,来自谷歌及其旗下 Chromium 开源开发团队终于揭开了全新语言 Dart 的神秘面纱,谷歌同时承诺该语言将为 Web 编码工作带来全新的构造方式。



Lars Bak——这位来自谷歌的工程师曾经凭借其 V8 JavaScript 解释程序颠覆了火狐浏览器的速度神话。如今根据他的说法,谷歌推出的全新 Web 开发语言 Dart 很可能与 Smalltalk 存在着种种相似之处。不过它似乎同时也在将 Web 发展引向一条与此前截然不同的另类道路。在今天于丹麦举行的 Web 开发者大会上,来自谷歌及其旗下 Chromium 开源开发团队的成员们终于揭开了全新语言 Dart 的神秘面纱,谷歌同时承诺该语言将为 Web 编码工作带来全新的构造方式。

要让 Dart 正常运作,我们需要部署一套全新的虚拟机,这种要求无形中使其与 Java、Adobe Flash 以及微软 Silverlight 天然形成了竞争关系。不过从体系结构角度而言,Dart 就目前来看将更多地依托于浏览器。

谷歌的计划是意欲将 Dart 虚拟机打造为对浏览器现有 JavaScript 引擎的一种补充。换句话说,它将把 Dart 代码加以重组并生成功能相同的 JavaScript 代码。尽管谷歌一直在回避讨论 Dart 嵌入全部浏览器之中的可能性,但在今早发布的博文中,Bak 明确指出在将 Dart 虚拟机整合到谷歌 Chrome 浏览器方面,“我们已经有计划探索这一方案。”

**终于,我们迎来了类**

Dart 语言存在的意义是为 JavaScript 提供辅

助功能,因为后者随着时间的推移,其在“开放式 Web”中的分布式应用程序方面已经无法胜任功能上的核心角色。比起结构化更彻底、面向对象特性更明确的 C# 及 Java,JavaScript 工作起来总会带来一些无法清除的累赘因素。而与诸如 Python、Ruby 以及 D 之类更为智能化的动态语言相比,JavaScript 又显得比较 嗦,无法迅速切入主题。

随着 Dart 语言首部教程的问世,Bak 及其架构团队以 JavaScript 的基础为起点,向其中加入了大量关键性 Java 要素,并通过对翻译程序潜在条件的改进大大优化了语言的整体简洁性。

对于 JavaScript 而言,实际上存在着一些能够使其更偏面向对象的类函数,而这些类函数所使用的正是关键字函数。我们可以对那些恰好具备属性的函数进行声明,而该函数将作为一种类函数嵌入其中。不过这种处理方式的问题在于,该类函数本身并没有得到重复调用,而是每次都以重新创建的形式出现,因此将该类函数以绑定的方式添加至函数原型中才是上佳的解决办法。当然,我们也可以对通用实例进行加值以实现对某种对象的声明,这可能需要利用常量对内部结构加以定义。接下来,从常量中选取一个,进而为函数赋值的做法也就使得该函数在某种意义上具备了变量值。

## Dart 之于 JavaScript 正如 C# 之于 C++ II

或者我们也可以对加值进行声明,并使用整个函数的结果为其赋值,这就相当于假设该函数是对象的惟一组成部分。

以上方法都不是特别理想。在 V8 以及 Mozilla 出品的 TraceMonkey 这些最新一代 JavaScript 引擎中,大部分的工作负载都是在将那些由不理想的指令所带来的“垃圾代码”转化为少些混乱、多些严谨的有序代码。

所以从理论角度出发,运行有 Dart 虚拟机的浏览器也许可以摆脱性能表现低下的噩梦,因为 JS 翻译程序所要处理的代码已经更为有序且易于理解。当然这只是理论。

Dart 重新将类的概念以 Java 程序员们最为熟悉的方式引入。其中几项调整使得定义工作更为简洁;举例来说,反向指针可能会被用于成员函数的声明(例如用 `Ball.throw (this.distance)` 代替 `Ball.throw (distance); this.distance = distance;`)。而接口也将为那些通用于多个类的属性及函数提供支持;一个类启用一个接口,这不正是大家一直以来所期望的结果么。

### 至今仍不明确:事件的触发

对那些始终关注 Dart 项目的 JS 开发人员来说,其中一大戏剧性结构变化也许会激起他们的广泛反对。JavaScript 函数通常是被粘贴于整个页面当中的,而且尽管其中某部分充当的是 `main()` 函数,大多数函数仍然是由浏览器事件所触发,包括 `window.onload`。而在 Dart 中,每个 `<script>` 元素本身都是个自成一体的环境。类成员以及变量在默认情况下都是公有的(在 Dart 语言中没有“`public`”这一关键字),尽管 Dart 的

本意并不是将公有属性推广到整个页面——而只是在脚本内部。

“在 JavaScript 当中,跨多个脚本标签的声明都在同一个命名空间内被结合在一起,” Sigmund Cherem 与 Vijay Menon 今早在文章中声称。“但在 Dart 中,同一个脚本标签中的代码无法直接访问被定义在另一个标签中的代码。如果某个脚本需要读取来自不同 URL 的代码,它必须通过 `#import` 方能实现。每个脚本标签要想运行都必须定义自己的 `main()` 接入点才行。

`main()` 函数是 `DOMContentLoaded` 事件的事件句柄。否则我们也不会看到早期的说明文档中提到任何关于 Dart 代码如何像 JavaScript 代码那样响应浏览器事件的声明。如果 Dart 根本没有做出任何变化,那么可以预见的是在 DOM 载入前不会有任何代码付诸运行;`main()` 函数也就可以直接留空,而让其它函数负责响应事件——这跟原先 JavaScript 所做的混蛋事完全没有区别。

不过想想就知道,上述计划将会为多线程处理带来相当简单而直接的处理方法:同一个页面可以拥有多个脚本,而虚拟机也能够使其并行运行…再次强调,这只是根据早期说明文档进行的理论层面分析。浏览器目前还不具备多线程翻译程序,而 Dart 也还没有明确指出其 JS 翻译程序中拥有如此猛料。综上所述,断言 Dart 能够调用这一目前尚不存在的功能实在是为时尚早。

Dart 已然具备了诸多优势,因此我们有足够的理由相信它可能将支持多款浏览器,甚至包括 IE。目前还待观察的是 Dart 将如何冲击浏览器上 HTML 5 应用程序或者是“本机应用程序”。■



# JavaScript大辩论:改进还是彻底放弃

谷歌的 Dart 语言将矛头直接指向了 JavaScript, 英特尔的 River Trail 则进一步推波助澜; 这场竞赛正因智能手机应用程序对开放 JavaScript 的需求而逐步升级。



无论大家如何看待 JavaScript, 都不会影响它当之无愧地成为现代计算中最重要编程语言队伍中的一员。这是惟一款能够在不添加任何额外插件的前提下随意运行于全部现代见面浏览器上的语言, JavaScript 可谓客户端页面领域的通用表达方式。

这种称号对于一种语言来说算是相当学生的负担了。当然, 也并不是每个人都对其投来善意的目光。随着网页应用程序的普及率及重要度不断提升, JavaScript 作为应对一切开发需求的解决方案, 所面对的压力也在与日俱增。

不可否认, JavaScript 在各方面的进步有目共睹。尤其是谷歌已经说服了各家浏览器供应商为 JavaScript 引擎的性能表现提供最大程度的支持, 该趋势也在顶级供应商之间造成了一次小规模军备竞赛。近年来, Mozilla 在其火狐浏览器上已经针对 JavaScript 引擎做出了显著的改善, 但也许微软才是整个事件中最大的受益者。微软得出的结论是 JavaScript 孱弱的性能表现对 IE 浏览器并无多大影响。如今微软甚至将 JavaScript 作为核心语言整合进了其赖以生存的 Windows 系统及 Office 办公软件当中。

但即使是改进如此巨大, 似乎仍然无法让每个人满意。由于开发人员对 JavaScript 的要求持

续激增, 其局限性也就被人为地放大化了。眼下谷歌刚刚放出消息, 这家向来倡导以浏览器应用程序代替桌面系统软件的企业长久以来一直在偷偷摸摸地着手开发一款名为 Dart 的全新语言。该语言预计在即将于丹麦举行的 Goto 大会上亮相, 其设计目的是彻底替代 JavaScript 这款“存在根本性缺陷”的语言。

说到这里, 我们需要思考一下了。Web 开发社区将会一如既往地支持 JavaScript 并努力将其打造成一流的开发平台, 而直接无视掉随之而来的局限性; 抑或是“做出一个艰难的决定”, 以良禽择木而栖的态度直接投向 Dart 的怀抱? 两种猜测似乎都有道理。

## 谷歌投下赌注

目前关于 Dart 语言的进一步信息还很有限, 估计这种情况会一直持续到 Goto 大会开幕。当下我们所了解到的大部分情况来自于一份意外泄露的备忘录, 该文由谷歌开发人员 Mark S. Miller 于 2010 年 11 月份所写, 原文题为“JavaScript 的未来”。不过仅从备忘录本身来看, 谷歌并非要打一场不是你死就是我亡的惨烈决战, 而是希望二种方案在未来的客户端 Web 领域得以共存。

在备忘录中, Miller 在 Dart 语言的推广方面透露了一套“风险系数极高”的方案选项。

## JavaScript 大辩论：实施改进还是彻底放弃 II

即其成功必须依赖浏览器厂商对 JavaScript 的“弃”无返顾以及谷歌自身的忽悠能力。而另一种方案——Miller 形容为“风险相对较低”——是对 JavaScript 进行改造以尽量解决其局限性。Miller 在文中最有价值的观点在于，他呼吁谷歌不要从上述两种方案中选择其一，而应该将二者同时解决。

而这正是谷歌过去一直在推行的思路。根据来自谷歌、一直效力于 Chrome 浏览器框架小组的 Alex Russell 的说法，在沸沸扬扬的争论之中（除了谷歌本身正是 Dojo 工具包的创始者之外），“谷歌明确承诺将争取改进 JavaScript 的薄弱环节，我们每位研究人员也一直在为此尽心竭力…而且说句实话，这也确实需要一定的助力。”

Russell 所主张的改变并不特别针对测试阶段的应用程序性能以及开发者生产力的提高。JavaScript 的开发模式已经经历了多年的发展，而目前所暴露出的问题是相对于其它比较先进的用法，JavaScript 在某些语法的处理上略显笨拙。举例来说，Russell 希望将类关键字加入到该语言中，以使面向对象的代码更加明确且清晰。他还坦言，如果能有一款 API 模块用于打造标准化机制的话，JavaScript 库的创建将会更加便捷。

### JavaScript 获得并行处理能力

对 JavaScript 的改良绝不是个通过纸上谈兵就能实现的简单计划，然而整个修正流程又需要在完全概念化的层面上进行讨论并模拟实施。Russell 曾作为谷歌方常驻 TC39 的代表，参与了由 ECMA（即欧洲计算机制造商协会）委员会负责的 ECMAScript 语言标准化工作，而这正是

JavaScript 的诞生基础。要对国际化标准进行改进实在不是易事，TC39 的成员们在 2008 年该委员会彻底放弃 ECMAScript 4 计划时彻底理解了这一点。

不过这种合作的过程还是非常重要的，因为谷歌并非唯一一家希望在客户端 Web 开发领域抢占先机的企业。举例来说，英特尔希望能使网页应用程序更好地利用现代多核心处理器设计，而 Dart 语言在这方面的表现还无从知晓。

本周，英特尔将自己的想法通过演示软件的形式派发给开发者们。这家老牌芯片制造商将其“全新 JavaScript 引擎”命名为 River Trail，并介绍称其为 JavaScript 带来了“并行处理的能力”，根据英特尔公司工程师 Stephan Herbut 的说法，这将使 JavaScript 成为一种高性能方案，“并足以应对像照片处理这样的大运算量应用程序。”除了并行处理能力，River Trail 还允许 JavaScript 充分利用现代芯片中的向量处理指令。

### 谷歌全面进军 Web 领域

讨论了这么多与 JavaScript 相关的话题，有一种现象值得思考，那就是谷歌为什么要在这趟浑水中抛出 Dart 语言？正如谷歌公司的 Miller 所言，“要想说服其它各家浏览器厂商团结起来支持同一款新语言简直比登天还难。”新语言总是层出不穷，但真正获得成功的却屈指可数。就拿谷歌最为得意的 Go 语言为例，自从于 2009 年推出以来除了在搜索巨头自家的办公室里插科打诨以外就没起过什么正经作用。（核子可乐译）

详细请查看：

<http://developer.51cto.com/art/201109/294390.htm> ■

# 微软能否撑起Silverlight的明天?

9月1日,微软 Silverlight 5 RC 发布,预计年末微软会正式推出 Silverlight 5 正式版。同时今年的 BUILD 开发者会议上,微软揭晓最新的 Silverlight 定位。微软创造 Silverlight 的目的,是为了促成一个跨平台跨浏览器的解决方案,并直接与 Flash 平台展开肉搏。然而这个目标似乎不再是那么重要了,恐怕当初的微软也不会想到今天的 Flash 平台本身也变得不是那么重要。一向自诩为跨平台的 Flash,却已经无法登陆乔帮主的 ios 战船,不得不说这是一种尴尬。Silverlight 与 Flash 成了难兄难弟,因此,微软果断地转移了自己的战略重点,投奔“真正的”跨平台跨浏览器技术,那就是 HTML 5,这是否意味着微软将抛弃 Silverlight,HTML 5 将从根本上替代 Silverlight? 在去年 PDC 大会微软的答案是:我们对 Silverlight 不抛弃不放弃。事隔一年回头看,这个亲妈对 Silverlight 还是很给力的,2011 年微软发布了 Silverlight 5 Beta 版,但在这个 HTML 5 的时代,微软能否撑起 Silverlight 的明天?

## Silverlight 的光芒

Flash VS HTML 5 的舆论或多或少掩盖了 Silverlight 的光芒,其实 Silverlight 在推出短短三年左右时间内就能触动企业开发者,这已经是了不起的成就。Silverlight 较之 WPF、Java Swing、Delphi 等 C/S 架构有更加轻量的运行环境与零维护的特点:使用轻量的 CLR Core 运行时环境,不依赖于客户端环境(无须安装体积庞大的 .Net Framework,这一点太棒了)。Silverlight 具有良好

的后台语言框架支持,这就是基于 .Net 高级语言的精简运行环境。Silverlight 依靠 .Net 平台,使用 C# 高级语言代替 JavaScript 脚本语言,实现了强大的客户端计算能力、支持多线程,继承了 WPF 丰富的样式、控件、特效与动画,更可控的浏览器适应性,更安全的沙箱模式,客户端嵌入式数据库等。另外 Silverlight 即较之传统应用有更友好的交互性,更酷的效果。支持完全面向用户体验的开发过程,其快速原型工具使需求与交付物更为明确,用户体验驱动开发,设计与编码分离。

微软企业副总裁 Scott Guthrie 承诺会发布 Silverlight 5 的一系列新特性,并宣布 Silverlight 5 将注重丰富的媒体体验与企业应用开发两大方面的改进。其中针对媒体方面的改进包括 GPU 硬解码、H.264 等 5 项功能,而针对企业应用方面的改进包括 64 位操作系统支持、IE 9 硬件加速、向量打印、文本清晰度、调用非托管代码等 30 多项功能,可以看出 Silverlight 已经逐渐将其未来重心转向企业级应用方面。

据调查新浪财经、腾讯、淘宝、口碑网等互联网企业已经尝试使用 Silverlight 开发交互性较强的商业应用,而一贯谨慎保守的大型金融公司陆续开始使用 Silverlight 来提高企业应用的用户体验,比如中国人寿(网上服务应用)、中国人保(商务智能应用)已经有相关应用,而像花旗银行、工商银行这样的大型银行也在尝试使用 Silverlight 来实现未来网上银行一些功能。■

查看详情:

<http://developer.51cto.com/art/201109/295038.htm>



# 数据库迁移之何去何从

随着世界联系越来越紧密,越来越智能,许多企业都希望其 IT 的使用也变得更加智慧。企业的数据量正在以前所未有的速度增长,业务对数据的依赖强度越来越高。如何让数据更高效帮助业务的发展,快速响应业务需求,在大量数据中及时提供决策分析,提升企业管理者对 IT 的管理效率,在当前的经济环境下让 IT 为企业带来更大经济效益,让 IT 帮助企业未来 5-10 年的业务发展——所有这些都对企业的数据库系统提出了更高的要求。此时,就出现了数据库的迁移。

为何要对数据库进行迁移?

回顾年初:联合可口可乐装瓶公司将数据库迁移到 IBM DB2 的事件。为什么可口可乐公司要进行迁移?51CTO 记者带着这一问题对 IBM 软件集团大中华区数据库和数据仓库销售总监何怡静女士进行了专访。何怡静女士表示现在的客户更多是考虑成本问题。确实,在数据库功能越来越强大的今天,客户对数据库的选择更多的考虑,如何用更小的成本获得更大的效率,当然如何降低成本,也是大家无论是客户还是合作伙伴面临的最巨大的挑战。

数据库间的迁移主要的一个原因是因为客户考虑的成本问题,举一个例子:维保问题。因为高额的维保,所以选择迁移到别家数据库产品,以降低成本。

再看另一个迁移例子:Digg 和 Reddit 宣布转向 Cassandra,因为 MySQL 对他们来说伸缩性不够了。一些人认为 MySQL+memchache 不再是事实上的伸缩解决方案了。

迁移都要考虑什么因素呢?

其实作为客户,对于他们使用的数据库,无论是出于成本的考虑,还是惯性的原因,亦或是感情的原因,大都是选择“不抛弃,不放弃”的原则。那么是什么原因使得这些忠实的“粉丝”选择了抛弃,选择了放弃呢?这里总结了以下几点原因。

## 一、成本

就像上文中提到的那样,成本是是否对数据库做迁移的一个根本性因素,谁能为客户带去更大的利益,客户可能会忽略在迁移中产生的成本,而选择“搬家”。我们还是以可口可乐公司为例,通过迁移,使用 DB2,可口可乐公司的存储需求减少了大约 40%;同时批处理时间也大幅减少了 65% 以上,从而提高了供应链的整体效率。

## 二、维保

维保的问题是连接着成本而来的,就像上文中何怡静女士为我们举的例子一样,也许客户从没有想过迁移,但是却因为高额的维保费用,从为公司减少不必要的开销的角度出发,最后决定从众多产品中选择出一个可以总体减少成本的数据库,而放弃之前的产品。

## 三、使用的普遍性

有时候选择使用某种数据库,也要看一看这种数据库的使用普遍性,如果拿到了一种数据库,却发现无从下手,那岂不是尴尬?

根据我们在卓越上输入关键词 SQL Server,查找结果有共 1,064 条,MySQL,共 198 条,Oracle,共 711 条,DB2,共 92 条。在当当上输入关键词 SQL Server,共搜到 519 个商品,MySQL,

## 数据库迁移之何去何从 II

共搜到313个商品, Oracle, 共搜到249个商品, DB2, 共搜到82个商品, 由此可见, 从图书上说, SQL Server的普遍性最广泛。

为了迁移的方便, 各家更是推出了适应别家迁移到自家的迁移工具, 微软有 MySQL 向 SQL Server 迁移工具 CTP, 不过这个工具只支持到 MySQL 的 4.1/5.0/5.1 版本, 不知对现在的 5.6 版本何时才能支持。

### 四、与时俱进

现在的数据形势是呈现海量的, 非结构化占据主要地位的, 数据库厂商是否能够 hold 住这个形势呢? 这也是客户是否会选择放弃该种数据库的一大原因。

DB2 的动作: IBM 表示已经在现有 DB2 产品中增加了对 hadoop 的支持, 在未来推出的第 10 版本中也会继续加强对海量数据和非结构化的支持。

SQL Server 的动作: 运行 SQL Server 的微软客户将通过 Hadoop 的引入获得真正的大数据处理能力。微软已经发布了早期代码, 让客户可以将这个 Java 架构接入到 SQL Server 2008 R2、SQL Server Parallel Data Warehouse 以及下一代微软数据库。

如果 hold 不住这一形式, 会怎么样呢?

情景一: 对不起, 我们离婚吧, 我爱上了别人。

Craigslist 采用 MongoDB 替代 MySQL

视觉中国的 NoSQL 之路: 从 MySQL 到 MongoDB

情景二: 亲爱的, 我们结婚吧, 我们会是最幸福的

新娘: Redis, 新郎: MySQL, 结婚地点: 新浪

新浪微博是 Redis 全球最大的用户, 在新浪有 200 多台物理机, 400 多个端口正在运行着 Redis, 有 +4G 的数据跑在 Redis 上来为微博用户提供服务。

在新浪 NoSQL 和 MySQL 在大多数情况下是结合使用的, 根据应用的特点选择合适存储方式。譬如: 关系型数据, 例如: 索引使用 MySQL 存储, 非关系数据库, 例如: 一些 K/V 需求的, 对并发要求比较高的放入 Redis 存储。

### 总结

究竟一个客户会在哪种情况, 选择将一个数据库从一个服务器移到另一个服务器上。这种迁移分两种情况, 一种是整个数据服务器全部迁移, 一种是只移其中的个别数据库。无论是哪种迁移, 是否都说明原有的数据库 hold 不住客户的需求呢? 总之, 现在看来迁移何去何从是客户说的算。

### ■ 开心一下





HTML 5 对比 Flash

### 成本 PK

HTML 不属于任何公司,谁都可以用文本编辑器使用这些技术,对开发者和用户都是免费的。但会增加开发者学习成本。

Flash 播放器是免费的,但开发工具很贵。不过 Flash 已经发展十年之久了,有成熟的技术社区和技术资料,没有再学习成本。

### 功能 PK

HTML 5 具有 2D 绘图能力,同时也可以自行执行多媒体技术、并拥有离线处理技术的新产品,具有 RIA 特性,同时还是 RIA 技术的平台、载体。

Flash 可以做很多很高级的特性。比如 Google 使用 Flash 实现 Gmail 的多文件上传。成千上万的 Flash 游戏在表现方面也是 HTML 5 望尘莫及的。

### 支持 PK

HTML 5 由浏览器解析执行,无需特别插件。已慢慢开始成了一个标准并被广泛的接收。但各大浏览器支持程度不同,普及还需要很长一段时间。

Flash 被流行的 iPhone、iPad 平台抛弃了,而且需要在其他的设备运行也要安装插件,但在 PC 端几乎全部支持。

想查看更多内容,请访问 51CTO 的 HTML 5 专题,链接:

<http://developer.51cto.com/developer/html5/>

HTML 5 是近十年来 Web 开发标准最巨大的飞跃。和以前的版本不同,HTML 5 并非仅仅用来表示 Web 内容,它的新使命是将 Web 带入一个成熟的应用平台,在 HTML 5 平台上视频、音频、图象、动画以及同电脑的交互都被标准化。

### HTML 5,下一代 WEB 开发标准

<http://developer.51cto.com/art/200907/133407.htm>